VEAS:
File systems, Cobalt, and other notes

First a definition: WAT

WAT |wat|

noun
(in Thailand, Cambodia, and Laos) a Buddhist monastery or temple.

verb, noun, adjective & exclamation informal
(in software or hardware) where logic and convention break down, such as when running on the EAS hardware and software; made famous in a 2012 CodeMash talk found at:
https://www.destroyallsoftware.com/talks/wat

Driver and OS status

- We're currently running Red Hat Enterprise Linux 6.2
  - this covers logins, the control system and infrastructure, and IO nodes
  - We are continuously patching where possible
    - holding back on anything that strongly impacts the toolchain, control system, or availability
    - 99% of packages are stock RHEL 6.2 ppc64 packages
  - everything is 64-bit
    - we will install 32-bit packages and applications if absolutely necessary
    - so far there haven't been any cross-compiling surprises
- Current driver is the 31 January 2012 driver
  - This driver is a version behind what LLNL is running
  - We're expecting a new driver by the end of April
  - We have neither source nor documentation, but again expect both April

Filesystems and Layout

- /soft
  - We'll get back to this
  - Mounted everywhere
  - NFS from NetApp filer
- /veas_home
  - Mounted everywhere
    - read-only on IO nodes
    - r/w on logins
  - NFS from NetApp filer
  - snapshot of fs taken each hour, night, and week
    - see /veas_home/.snapshot/
- /veas-fs0
  - mounted everywhere
  - GPFS 3.4.0-6
  - served from 4 DDN sfa1000ke controllers over 16 40 GB/s adapters

Other important notes

- There are no backups.
  - Yes, we just said there are snapshots of /veas_home
    - they are on the NetApp appliance and are toast if disk gets full or something happens to the hardware
  - We're not keeping tape backups
  - We don't offer an archival facility at this time
  - We are planning to migrate data when production file systems are ready
- Bandwidth is far less than you'll see in production
  - NFS file systems are coming over a10 GigE interface
  - There aren't that many spindles behind the filesystems

A moment of WAT

- Yes, we're aware that the system is periodically eating stdout and stderr
  - right now it looks like the cios daemon gets backed up (but does not complain), the IO node kernel panics, the panic propagates to the IO node designated as backup
  - we don't get a RAS event or any useful control system message
  - there *is* a watcher script being setup to set a reservation when the panic is seen
    - likely by Wednesday it'll be set
  - IBM knows about the behavior but we're still waiting on a fix

Cobalt

- Very similar to /P with caveats
  - modes are different
    - -c{1,2,4,8,16,32,64} sets ranks per core
  - -n gives you nodes
  - --proccount gives you total processes
  - custom kernels are not yet supported due to control system limitations
  - the cqsub and cqstat commands are going the way of the dodo
    - use qstat and qsub
- Script mode is the same
  - block starts off booted
  - cobalt-subrun does not work
  - see wiki for details (use runjobs in scripts - only in scripts)

Block naming and Cobalt

- Block names follow logical names, not hardware names
- Why?
  - 32 character limit on block names
  - Mira allows many more degrees of freedom in block configurations
  - Allows us to state which midplanes, and by extension which hardware is in use in a given location when the hardware locations make little sense
  - Makes sub-block setup easier
- One rack has the topology 4x4x4x8x2
- One midplane is 4x4x4x4x2

Decoding Block Names

- LOC-CCCCC-XXXXX-[T]-[PPPP]-SIZE
- LOC = location identifier, like ANL, CHR, VES, CET, MIR, EAS can be up to 7 characters.
- CCCCC = The bottom right front corner as described as a set of 5-dimensional coordinates ABCDE. This corresponds to the node location of the node of rank 0 in a ABCDE-type mapping scheme (node 0).
- XXXXX = The top left rear corner of the block in each dimension (node n-1).
- T = an optional identifier indicating which dimensions are Mesh and which are torus. This is a bitmasked value (0 = toruS, 1 = mesh). No value implies the maximum number of torus dimensions for that block
- PPPP = indicator of passthrough extents in each dimension. This will have a value of 0, 1, or 2.
- SIZE = The overall size in nodes of the block. This should correspond to the product of the extents.

Block name example

- A sample logical address could be: MIR-04C00-48FF2-7-2048
- Think LOC-CCCCC-XXXXX-[T]-[PPPP]-SIZE

- This corresponds to:
  - one midplane in the A dimension, first midplane in A
  - one midplane in the B dimension, starting at the second midplane (row 1, to be exact)
  - one midplane in the C dimension,
  - Four midplanes in the D dimension,
  - A,B,C dimensions are mesh, D is a torus
  - 2048 Nodes

- Old style it might be: MIR-R14-R15-2048

Resource Isolation

- Resource sets where you are the only one on our resources.
  - >= 512 (midplane or above): The IONs, computes and blocks are all yours within that block)
- You are shared when:
  - >= 256 : you share IONs
  - > 128: you share the block with other users (i.e. 2 64s may have different users). May see traffic on interconnect from other users (I think, I figure this has to be the case for traffic to the ION through J06 and J11)
- You are always the only thing on the compute node's compute cores and memory

/soft layout and finding things

- We’re trying to reorganize /soft to make things easier to follow
  - Arrangements will be by function, ie: compilers in /soft/compilers, performance tools in /soft/perftools, softenv and modules in /soft/environment.
- softenv keys should be authoritative
- front end software (editors, X, games) is installed in RHEL’s default locations